AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) In a computer system that supports a visual user interface development tool, a method of centrally managing user interface state information for the visual user interface development tool such that behavior for one or more user interface components or the visual user interface development tool itself may be defined dynamically at development time, the method comprising acts of:

creating a transparent adorner window overlaying a design space of the visual user interface development tool, the transparent adorner window being adapted to intercept messages of the visual user interface development tool;

~~receiving~~ intercepting a message generated within the visual ~~a visual~~ user interface development tool during the development of one or more user interface components;

sending the message to a centralized extensible behavior stack to check for one or more behaviors to use in processing the message, wherein a behavior describes the manner in which the user interface components or the visual user interface development tool processes the message resulting in a user interface activity dependent upon the behavior such that the functionality of the user interface components or the visual user interface development tool is defined by the behavior;

checking the centralized extensible behavior stack containing currently available behaviors for processing messages to determine if a behavior is available to process the message; and

if a behavior is available on the centralized extensible behavior stack, then passing the message to the available behavior for processing thereby resulting in a dynamically defined ~~behavior~~ functionality and appearance of the interface.

2. (Previously presented) A method as recited in claim 1, wherein the behavior is available on the centralized extensible behavior stack, and wherein the behavior is associated with the visual user interface development tool, as opposed to an individual user interface component within the visual user interface development tool.

3. (Previously presented) A method as recited in claim 1, wherein the behavior is available on the centralized extensible behavior stack, and wherein the behavior is associated with an individual user interface component within the visual user interface development tool, as opposed to the visual user interface development tool itself.

4. (Original) A method as recited in claim 3, wherein the individual user interface component comprises a third party component developed separately from the visual user interface development tool.

5. (Original) A method as recited in claim 3, wherein the behavior comprises asking the individual user interface component for any glyphs that are part of the individual user interface component.

6. (Previously presented) A method as recited in claim 1, further comprising acts of:

receiving the behavior from a component within the visual user interface development tool during development time; and

pushing the behavior on the centralized extensible behavior stack.

7. (Previously presented) A method as recited in claim 1, wherein no behavior is available on the centralized extensible behavior stack for processing the message, the method further comprising an acts of:

checking for a successfully hit tested glyph with a corresponding glyph behavior for the message; and

if available, passing the message to the glyph behavior of the successfully hit tested glyph.

8. (Original) A method as recited in claim 1, further comprising an act of receiving one or more glyphs with corresponding glyph behavior from a component within the visual user interface development tool during development time, wherein each of the one or more glyphs is capable of hit testing and painting itself.

9. (Currently Amended) For a computer system that supports a visual user interface development tool, a computer program product comprising one or more computer readable storage media storing computer executable instructions that implement a method of centrally managing user interface state information for the visual user interface development tool such that behavior for one or more user interface components or the visual user interface development tool itself may be defined dynamically at development time, the method comprising acts of:

creating a transparent adorner window overlaying a design space of the visual user interface development tool, the transparent adorner window being adapted to intercept messages of the visual user interface development tool;

~~receiving~~ intercepting a message generated within the visual user interface development tool during the development of one or more user interface components;

sending the message to a centralized extensible behavior stack to check for one or more behaviors to use in processing the message, wherein a behavior describes the manner in which the user interface components or the visual user interface development tool processes the message resulting in a user interface activity dependent upon the behavior such that the functionality of the user interface components or the visual user interface development tool is defined by the behavior;

checking the centralized extensible behavior stack containing currently available behaviors for processing messages to determine if a behavior is available to process the message; and

if a behavior is available on the centralized extensible behavior stack, then passing the message to the available behavior for processing thereby resulting in a dynamically defined functionality and appearance if the ~~behavior of an~~ interface.

10. (Previously presented) A computer program product as recited in claim 9, wherein the behavior is available on the centralized extensible behavior stack, and wherein the behavior is associated with an individual user interface component within the visual user interface development tool, as opposed to the visual user interface development tool itself.

11. (Previously presented) A computer program product as recited in claim 9, the method further comprising acts of:

receiving the behavior from a component within the visual user interface development tool during development time; and

pushing the behavior on the centralized extensible behavior stack.

12. (Original) A computer program product as recited in claim 11, wherein the behavior corresponds to a particular action either being performed or to be performed on a user interface component within the visual user interface development tool, the method further comprising an act of popping the behavior off the centralized behavior stack when the particular action is completed.

13. (Previously presented) A computer program product as recited in claim 12, wherein the centralized extensible behavior stack enforces the existence of a single state for the particular action.

14. (Previously presented) A computer program product as recited in claim 9, wherein no behavior is available on the centralized extensible behavior stack for processing the message, the method further comprising an acts of:

checking for a successfully hit tested glyph with a corresponding glyph behavior for the message; and

if available, passing the message to the glyph behavior of the successfully hit tested glyph.

15. (Original) A computer program product as recited in claim 9, the method further comprising an act of receiving one or more glyphs with corresponding glyph behavior from a component within the visual user interface development tool during development time, wherein each of the one or more glyphs is capable of hit testing and painting itself.

16. (Canceled)

17. (Currently Amended) A computer program product as recited in ~~claim 16~~claim 1, wherein the one or more glyphs are organized into one or more adorner layers.

18. (Original) A computer program product as recited in claim 17, the method further comprising an act of disabling at least one of the one or more adorner layers.

19. (Currently Amended) In a computer system that supports a visual user interface development tool, a method of centrally managing one or more behaviors that are dynamically defined at development time for a component within the visual user interface development tool or for the visual user interface development tool itself, the method comprising steps for:

catching a message generated in response to user input within the visual user interface development tool during the development of one or more user interface components;

routing the message to a centralized and extensible behavior store that contains currently available behaviors to determine any behaviors that are available for use in processing the message, wherein a behavior describes the manner in which the user interface components or the visual user interface development tool processes the message resulting in a user interface activity dependent upon the behavior such that the functionality of the user interface components or the visual user interface development tool is defined by the behavior;

determining if the centralized and extensible behavior store includes one or more behaviors for processing the message; and

if a behavior is included within the centralized and extensible behavior store, then using the behavior to process the message thereby resulting in a dynamically defined functionality and appearance ~~behavior~~ of an interface.

20. (Original) A method as recited in claim 19, wherein the behavior is available on the centralized behavior stack, and wherein the behavior is associated with an individual user interface component within the visual user interface development tool, as opposed to the visual user interface development tool itself.

21. (Original) A method as recited in claim 19, further comprising:

an act of receiving the behavior from a component within the visual user interface development tool during development time; and

a step for adding the behavior to the extensible behavior store.

22. (Original) A method as recited in claim 19, wherein no behavior is available on the centralized behavior stack for processing the message, the method further comprising steps for:

determining if a successfully hit tested glyph with a corresponding glyph behavior exists for the message; and

if the successfully hit test glyph exists, using the corresponding glyph behavior to process the message.

23. (Original) A method as recited in claim 22, wherein no successfully hit test glyph with corresponding glyph behavior is available for the message.

24. (Original) A method as recited in claim 19, further comprising an act of receiving one or more glyphs with corresponding glyph behavior from a component within the visual user interface development tool during development time, wherein each of the one or more glyphs is capable of hit testing and painting itself.

25. (Original) A method as recited in claim 19, wherein the message comprises one of a user event, a mouse message, and a keyboard message.

26. (Original) A method as recited in claim 19, wherein the centralized and extensible behavior store contains all currently available behaviors.

27. (Currently Amended) For a computer system that supports a visual user interface development tool, a computer program product comprising one or more computer readable media carrying computer executable instructions that implement a method of centrally managing one or more behaviors that are dynamically defined at development time for component within the visual user interface development tool or for the visual user interface development tool itself, the method comprising steps for:

catching a message generated in response to user input within the visual user interface development tool during the development of one or more user interface components;

routing the message to a centralized and extensible behavior store that contains currently available behaviors to determine any behaviors that are available for use in processing the message, wherein a behavior describes the manner in which the user interface components or the visual user interface development tool processes the message resulting in a user interface activity dependent upon the behavior such that the functionality of the user interface components or the visual user interface development tool is defined by the behavior;

determining if the centralized and extensible behavior store includes one or more behaviors for processing the message; and

if a behavior is included within the centralized and extensible behavior store, then using the behavior to process the message thereby resulting in a dynamically defined functionality and appearance ~~behavior~~ of an interface.

28. (Original) A computer program product as recited in claim 27, wherein the behavior is available on the centralized behavior stack, and wherein the behavior is associated with an individual user interface component within the visual user interface development tool, as opposed to the visual user interface development tool itself.

29. (Original) A computer program product as recited in claim 27, the method further comprising:

an act of receiving the behavior from a component within the visual user interface development tool during development time; and

a step for adding the behavior to the extensible behavior store.

30. (Original) A computer program product as recited in claim 27, wherein no behavior is available on the centralized behavior stack for processing the message, the method further comprising steps for:

determining if a successfully hit tested glyph with a corresponding glyph behavior exists for the message; and

if the successfully hit test glyph exists, using the corresponding glyph behavior to process the message.

31. (Original) A computer program product as recited in claim 27, wherein the behavior defines a new custom behavior previously unavailable within the visual user interface designer.

32. (Original) A computer program product as recited in claim 27, the method further comprising an act of receiving one or more glyphs with corresponding glyph behavior from a component within the visual user interface development tool during development time, wherein each of the one or more glyphs is capable of hit testing and painting itself.

33. (Original) A computer program product as recited in claim 33, wherein the one or more glyphs are organized into one or more adorner layers.

34. (Original) A computer program product as recited in claim 33, the method further comprising an act of disabling at least one of the one or more adorner layers.

35. (Original) A computer program product as recited in claim 32, wherein the one or more glyphs comprise at least one custom glyph for the component.

36. (Original) A computer program product as recited in claim 32, wherein the message corresponds to at least one of a hit test message and a paint message.

37. (Currently Amended) A computer program product comprising one or more computer readable media carrying computer executable instructions that centralizes component behavior for a visual user interface development tool and permits a component to define at development time one or more custom behaviors that are specific to the component itself or applicable the visual user interface development tool, the computer executable instructions comprising:

an extensible behavior stack that contains one or more development time specified behaviors for the visual user interface development tool or a component within the visual user interface development tool, wherein a behavior describes the manner in which the user interface components or the visual user interface development tool processes the message resulting in a user interface activity dependent upon the behavior such that the functionality of the user interface components or the visual user interface development tool is defined by the behavior ~~wherein a behavior describes how the user interface development tool or component within the visual user interface development tool interacts with a user~~;

a extensible collection of one or more adorners, each containing one or more development time specified glyphs capable of hit testing and painting themselves, wherein at least one of the one or more glyphs includes a reference to a glyph behavior to invoke when a successful hit test has been determined; and

a message router that routes one or more received messages generated in response to user input within a visual user the visual interface development tool to either the extensible behavior stack or the extensible collection of one or more adorners.

38. (Original) A computer program product as recited in claim 37, the computer executable instructions further comprising an adorner window that intercepts one or more messages directed to the visual user interface development tool.

39. (Original) A computer program product as recited in claim 37, wherein the message router routes a received user event message, a received mouse message, or a received keyboard message to the extensible behavior stack.

40. (Original) A computer program product as recited in claim 37, wherein the message router routes a received a received hit test message or a received paint message to the extensible collection of one or more adorners.

41. (Original) A computer program product as recited in claim 37, wherein the one or more adorners organize the one or more development time specified glyphs into layers which can be independently disabled and enabled.

42. (Original) A computer program product as recited in claim 37, wherein the component within the visual user interface development tool comprises a third party component developed separately from the visual user interface development tool.

43. (Previously Presented) A method as recited in claim 1, wherein the dynamically defined behavior is directly related to at least one functionality of the interface component selected from the group comprising: dragging an object, resizing an object, and selecting an object.